

# A Mixed-Signal Early Vision Chip with Embedded Image and Programming Memories and Digital I/O

G. Liñán-Cembrano, A. Rodríguez-Vázquez, R. Domínguez-Castro and S. Espejo.  
Instituto de Microelectrónica de Sevilla. IMSE-CNM-CSIC  
Avda. Reina Mercedes s/n 41012 Sevilla (SPAIN)

## ABSTRACT<sup>A</sup>

From a system level perspective, this paper presents a  $128 \times 128$  flexible and reconfigurable Focal-Plane Analog Programmable Array Processor, which has been designed as a single chip in a  $0.35\mu\text{m}$  standard digital 1P-5M CMOS technology. The core processing array has been designed to achieve high-speed of operation and large-enough accuracy ( $\sim 7\text{bit}$ ) with low power consumption. The chip includes on-chip program memory to allow for the execution of complex, sequential and/or bifurcation flow image processing algorithms. It also includes the structures and circuits needed to guarantee its embedding into conventional digital hosting systems: external data interchange and control are completely digital. The chip contains close to four million transistors, 90% of them working in analog mode. The chip features up to 330GOPs (Giga Operations per second), and uses the power supply (180GOP/Joule) and the silicon area ( $3.8\text{ GOPS/mm}^2$ ) efficiently, as it is able to maintain VGA processing throughputs of 100Frames/s with about 15 basic image processing tasks on each frame.

## 1. INTRODUCTION

Already in 1997, forecasts regarding the next wave of InfoTech Innovation <sup>1</sup> anticipated that high-performance sensors would shape the first decade of the 3<sup>rd</sup> millennium. Thus, while in the 1980s innovations were focused on creating processor-based computer “intelligences”, and in the 1990s on networking those intelligences together with laser-enabled bandwidth, innovations during this decade will most likely be focused on adding *sensing* and *actuating* “organs” to these devices and networks.

In this area we have a lot to learn from nature. The close integration, of interacting sensing, processing and acting substructures featured by natural beings provides an endless source of inspiration for these new generations of sensorial intelligences <sup>2</sup>. Exploiting this source of inspiration may lead to revolutionary changes in the conception and implementation of these systems.

Thus, the extensive usage of digital processing in today’s conventional architectures <sup>b</sup> is foreseen to be complemented in the future with an increased usage of *parallel analog* processors capable of operating concurrently and in close interaction with the sensory circuits. The final target is to realize complete sensory/processing (and actuating) systems on a single chip through the smart synergy of sensors, analog processing and digital processing structures.

In this sense, during the last few years significant advances have been made regarding the implementation of *Vision Chips*; i.e. chips which are capable of acquiring images and processing them using circuits embedded in the same silicon substrate in which the image is captured – called *focal-plane processing*.

The design of these chips can be undertaken following two alternative approaches:

- Pick up a specific task and its model and implement it on silicon. This is the usual way, leading to very useful, task-specific smart sensors <sup>3, 4</sup>.

---

a.This work has been partially funded by DICTAM-IST 1999-19007.

b.These conventional architectures employ analog blocks only at the front-end sections, while all processing is realized in the digital domain. The whole sequence of operations is: Sensing – A/D Conversion – Digital-Processing – D/A Conversion)

- Make *general-purpose* mixed-signal image processing devices<sup>5</sup>. That is, devices which, through *programming*, can be employed to realize a myriad of image processing tasks, similar to the possibility featured by the ubiquitous Von Neumann digital processor.

The chip reported at system-level in this paper, called ACE16k, belongs to this second group. It has a Single-Instruction-Multiple-Data architecture (SIMD) and includes some of the most relevant features of the Cellular Nonlinear/Neural Network Universal Machine (CNNUM) paradigm<sup>6</sup>. Other chips, also belonging to this group, have been reported in the last years<sup>7-17</sup>, etc. Some of them do not have sensory capabilities, they are just conceived as programmable *co-processors*; others are only designed for *black-and-white* image processing; and others are designed for very large density, but feature very small analog *accuracy* in the internal processing.

The chip presented at system level in this paper outperforms all these previous implementations in terms of functionality, flexibility for reconfiguration, programmability features, input/output interface, speed of operation, accuracy of the internal analog processing, speed of internal operation and power consumption. Some features of this chip are summarized in Table.1.

## 2. ARCHITECTURE

ACE16k follows the SIMD-CNNUM paradigm. As shown in Fig. 1, Single Instruction Multiple Data systems consist of an array of identical Processing Elements (PE) which execute the same instructions at the same time. Instructions are executed on data which are *locally* defined, at the PE level, while the instruction sequence is issued by a control unit shared by all the PEs in the array. Most commonly, the communication network among PEs is restricted to the *nearest neighbors*.

Table 1: ACE16k Characteristics

Technology	ST Microelectronics 0.35 $\mu\text{m}$ 5M-1P
Design Style	Full Custom (Analog Core) and Standard Cells (Digital I/O block)
Package	Ceramic QFP144
# of Cells	16384 (128 x 128 Array)
# of Transistors	3,748,170
# of Transistors per cell	198
Cell Size	75.7 $\mu\text{m}$ x 73.3 $\mu\text{m}$
Cell Density	180 cells/ $\text{mm}^2$
Pixel Signal Swing	[0.6, 1.4]V (Programmable)
Weight Signal Swing	[2.15, 2.95]V (Programmable)
Accuracy of Analog Processing Blocks	$\sim 1\%$
Time-Constant -linear. convol.-	$\sim 160\text{ns}$
Time-Constant -CT Dynamics-	$\sim 0.8\mu\text{s}$
I/O Master Clock	32 MHz
Power Supply	3.3V $\pm 10\%$
Power / Speed / Area Figures	$0.33 \times 10^{12}\text{OPS}$ , $0.18 \times 10^{12}\text{OP/J}$ and $3.8 \times 10^9\text{ OPS}/\text{mm}^2$
# of Analog Instructions in mem.	32
# of Digital Instructions in mem.	64 x 64 Configurations
Die Size	11885.0 $\mu\text{m}$ x 12230 $\mu\text{m}$

In the case of *low-level*<sup>c</sup> image processing, where the same operation sequence is applied to all the processors in the array, the most straightforward mapping between images and chips consists of using one processing element per pixel, thus providing a very compact, and efficient way of defining algorithms.

ACE16k implements most of the relevant functional features of the CNUM<sup>6</sup> paradigm, namely:

- Non-linear dynamic coupling among the elementary processing units, also called *cells*,
- Local, distributed memories for storage of intermediate images,
- Local, distributed logical processing,
- Incorporation of *global*, chip-level structures for storage of *instructions* and control of algorithmic processing flows<sup>19</sup>.

Thus, ACE16k is capable of operating as a flexible, user-programmable algorithmic processor; a kind of *visual microprocessor*<sup>5</sup>. At the hardware level, the instruction set of such a microprocessor includes setting the values for the strengths of the cell interconnections, called interconnection *templates* which define the actual *low-level* image processing task to be executed; also implies reconfiguring the interconnection topology of the structures incorporated at cell level, thus controlling data flows and basic circuit configuration; and arranging local analog and digital operations between locally-stored images, which allows for the execution of both pixel-wise binary inversion in black and white images, for instance, and image-wise operations - gray-scale combinations of two images by linear arithmetic operations.

The architecture of the system is depicted in Fig. 2(a). Here, various functional substructures are identified:

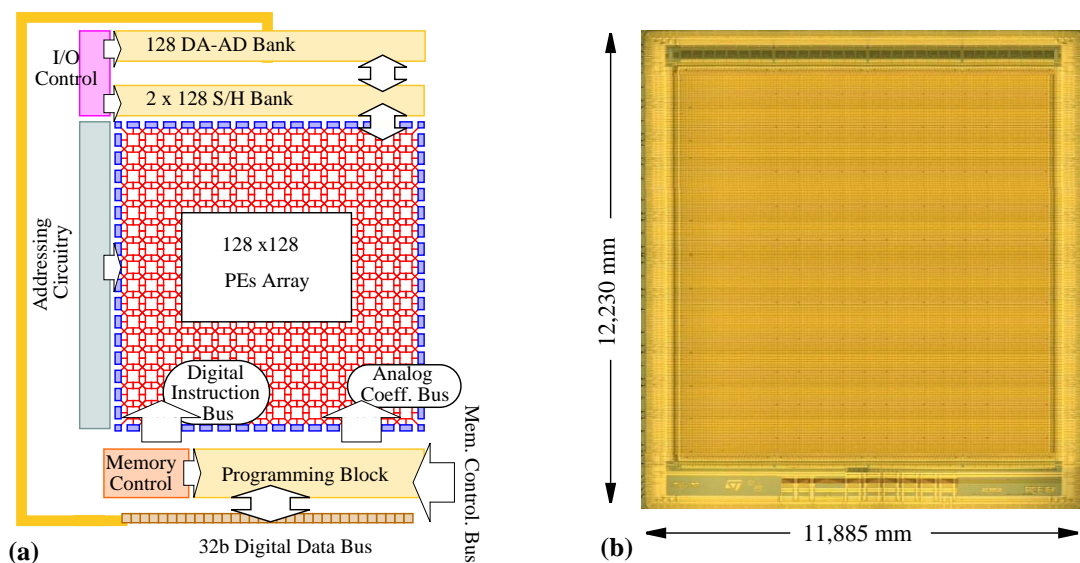


Figure 2: Architecture and microphotograph of ACE16k.

c. This refers to the processing realized at the early stages of the flow<sup>18</sup>, where the amount of data to process has very large dimensionality,  $N \times M$ , where  $N$  is the number of rows in the array of pixels, and  $M$  is the corresponding number of columns. This low-level processing stage is critical for reducing the dimensionality of the data for subsequent processing stages.

- A sensory/processing core, which consists of an array of identical  $128 \times 128$  PEs. These PEs have embedded circuit structures for optical sensing, programmable analog processing (designed for around 7bit accuracy), programmable binary processing, local memory and signal flow reconfiguration.
- A ring of *border* cells used to establish spatial boundary conditions for image processing, and several buffers driving analog and digital signals to the array.
- A programming block, which contains several SRAM digital memories used to store the algorithms to be executed by the chip.
- A block for I/O image flow control and format conversion<sup>d</sup>. 128 Digital to Analog (DAC) and Analog to Digital (ADC) converters, one per column, which constitute a digital I/O port for images.
- Digital blocks for automatic addressing of rows and columns during image I/O processes.

The chip uses a 32-bit bidirectional data bus for image communication purposes, and several address buses for the different blocks within the programming memory. The I/O interface follows very simple hand-shaking protocols. Table.1 summarizes the main characteristics of the prototype.

ACE16k is conceived to be used in two alternative ways. First, whenever the images to be processed are directly acquired by the optical input module of the chip<sup>5</sup>; and second, as a conventional image co-processor working in parallel with a digital hosting system which provides and receives the images in electrical form.

### 3. PROGRAMMING BLOCK

The programming block, illustrated in Fig. 3, provides the algorithmic capability of ACE16k. It is, basically, a set of 8 SRAM memory blocks with different type of contents. For instance, these contents vary from sets of digital signals, vectors, defining the algorithms to be executed –what we call “digital instructions”–, to sets of cell-to-cell interaction coefficients and reference levels to be applied to the array of processors –what we call “analog instructions”.

Two operating modes, namely the *programming* and the *operation* mode, are involved in the usage of ACE16k.

In the programming mode, each of the 8 SRAM blocks is independently accessed through the global data bus in order to be written. So, in this phase, the user defines the vision processing algorithm to be executed afterwards. On the other hand, in operation mode, the contents of the programming memory are selected through different address buses, and

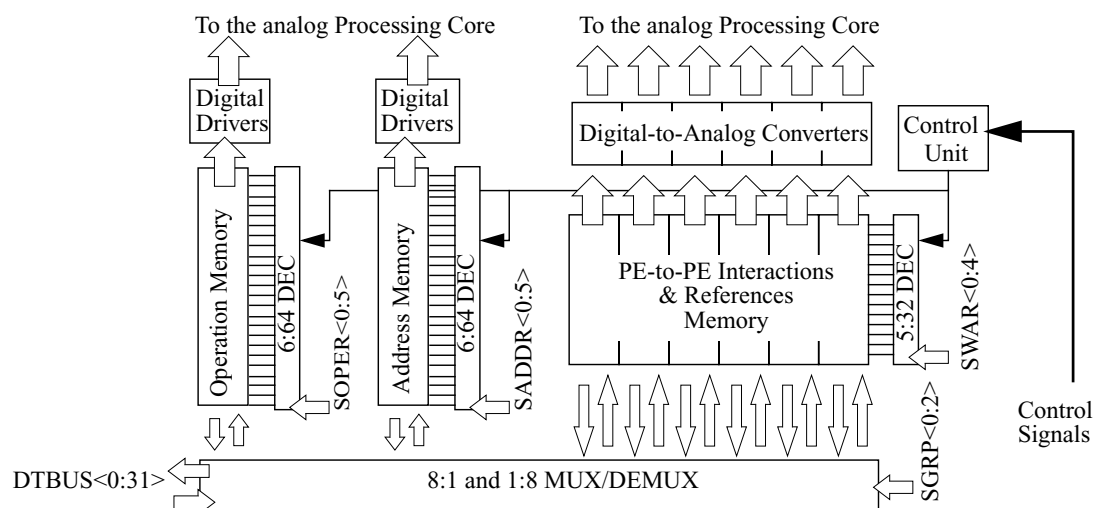


Figure 3: Block diagram of the programming block

d.From digital to analog and vice versa.

transmitted to the cell array to perform the specific image processing task programmed by the user in the previous programming phase.

Regarding their contents, different blocks in the programming memory can be classified into three categories. Two of them, Operation Memory and Address Memory, are used to store *digital instructions*. Each of these blocks is designed to store 64 words of 32-bit. A digital instruction is defined as a 64-bit digital vector that controls the configuration of the internal circuitry of the chip. It comprises a word from the operation memory – 32-bit – and another one from the addresses memory – 32-bit. The third group, the interactions and references memory, is used to store PE-to-PE interaction coefficients and some reference levels. This group comprises six identical SRAM blocks, each of them designed to store 32 words of 32-bit. Since analog coefficients are defined as 8-bit digital words, each block in this group stores 32 sets of 4 analog values. An *analog instruction* comprises 24, i.e.  $6 \times 4$ , analog values that are transmitted in parallel to the processing core by means of a bank of 24 digital-to-analog converters.

#### 4. ANALOG CORE

Fig. 4(a) shows the block diagram of the PE in ACE16k. Arrows indicate information flows. This diagram contains 8 building blocks which are interconnected through the so-called *ACE-BUS*. Data transferences are always carried out in the same way. A given block, the data source, writes the desired data to the ACE-BUS while another, the data destination, acquires this information from the ACE-BUS. Since PEs are in fact analog entities, this bus consists of just a single wire.

Beside the analog processing kernel, which will be described afterwards, the PE contains the following functional blocks:

- 1) Analog 1 Random Access Memory, LAM, with capability for storing 8 gray-scale pixel values with an equivalent resolution of 8 bits.
- 2) Local Logic Unit, consisting of a fully programmable<sup>e</sup> two-input one-output Boolean operator.
- 3) Multimode optical sensor.
- 4) An Address Event Downloading Module, which allows the chip to download, sequentially, the location of active pixels.
- 5) A resistive grid module which allows for the execution of continuous-time diffusion in a resistive-grid-like manner.

##### 4.1. Image Processing Kernel.

Each PE updates its state driven by the cells located within its neighborhood. Since the strengths of the interaction pattern must be programmable, a bank of analog multipliers is used to implement these interactions. These analog multipliers, designed by using a one transistor technique<sup>11</sup>, are driven by voltages at both inputs (the signal input and the scaling input) and provide a current at the output. The bank of multipliers, depicted at the conceptual level in Fig. 4(b), is driven by three different pixel values,  $P_A$ ,  $P_B$  and  $P_C$  so that the current which flows into the PE is expressed as,

$$I_{tot} = \mathbf{A} \bullet \mathbf{P}_A + b \cdot P_B + c \cdot P_C + z + I_{off} \quad (1)$$

where the  $\mathbf{A}$  and  $\mathbf{P}_A$  are defined as,

$$\mathbf{A} = \begin{bmatrix} a_{br} & a_{bc} & a_{bl} \\ a_{cr} & a_{cc} & a_{cl} \\ a_{tr} & a_{tc} & a_{tl} \end{bmatrix} \quad \mathbf{P}_A = \begin{bmatrix} P_{A_{tl}} & P_{A_{tc}} & P_{A_{tr}} \\ P_{A_{cl}} & P_{A_{cc}} & P_{A_{cr}} \\ P_{A_{bl}} & P_{A_{bc}} & P_{A_{br}} \end{bmatrix} \quad (2)$$

the operator  $(\bullet)$  accounts for the convolution product of those matrices<sup>f</sup>, and  $I_{off}$  is a spurious offset term produced by the one transistor multiplier.

e. The truth table of the logic function to be executed is part of the digital instruction.

f. That is, adding the results of multiplying each element in one operand by the element in the same position in the other.

The currents generated by the multipliers are collected by the input block of the PE, also in Fig. 4(c), and are sent to a very simple current processing block. The offset term generated by the multiplier is subtracted by using a high accuracy current memory block based on an  $s^3I$  memorization scheme -see Fig. 4(c). Afterwards, the actual signal current,

$$I_{in} = \mathbf{A} \cdot \mathbf{P}_A + b \cdot P_B + c \cdot P_C + z \quad (3)$$

can be either directly steered to the ACE-BUS or sent to the input of a current comparator<sup>20</sup>, whose output is connected to the ACE-BUS through an analog switch – see Fig. 4(c).

Two different situations occur depending upon whether this latter switch is ON or OFF.

- If the switch is ON, a voltage is delivered to the ACE-BUS which corresponds to the sign of  $I_{in}$ ; i.e. to the sign of the convolution operation,

$$\text{sign}(\mathbf{A} \cdot \mathbf{P}_A + b \cdot P_B + c \cdot P_C + z) \quad (4)$$

In this case the output is a black-and-white pixel value.

- If the switch is OFF, the analog current  $I_{in}$  is routed to one of the capacitors associated to the pixels and the output is a gray-scale pixel value.

In the latter case above, the specific capacitor to which  $I_{in}$  is routed is selected by the user through the activation of some bits in the digital instruction. By so doing, the evolution of the PE is described by a state equation whose actual expression depends on the selected integrating capacitor. Therefore, different kinds of processing kernels will be available.

Consider for instance a Sobel operator<sup>18</sup>. The convolution matrix is then defined in  $\mathbf{A}$ ; the image is loaded into  $P_A$ ; the following values are set:  $c = z = 0$ , and  $b = -1$ ; and the signal current is routed to  $C_B$ . Hence, the equivalent state equation obtained for each PE is,

$$C_B \frac{dP_B}{dt} = -P_B + \mathbf{A} \cdot \mathbf{P}_A \quad (5)$$

whose steady state is  $P_B = \mathbf{A} \cdot \mathbf{P}_A$ , as corresponds to the desired output.

Consider now that the capacitor which receives the input current is  $C_A$ . Then the cells are dynamically coupled and CNN spatio-temporal operations<sup>5, 19</sup> are realized. Finally if the current is routed to  $C_A$ ; that all but the central entries of matrix  $\mathbf{A}$  in (2) are null; and that this central entry is  $a_{cc} = -1$ . The steady state solution is then,

$$P_A = b \cdot P_B + c \cdot P_C + z \quad (6)$$

which corresponds to the realization of gray-scale image-wise arithmetic operations.

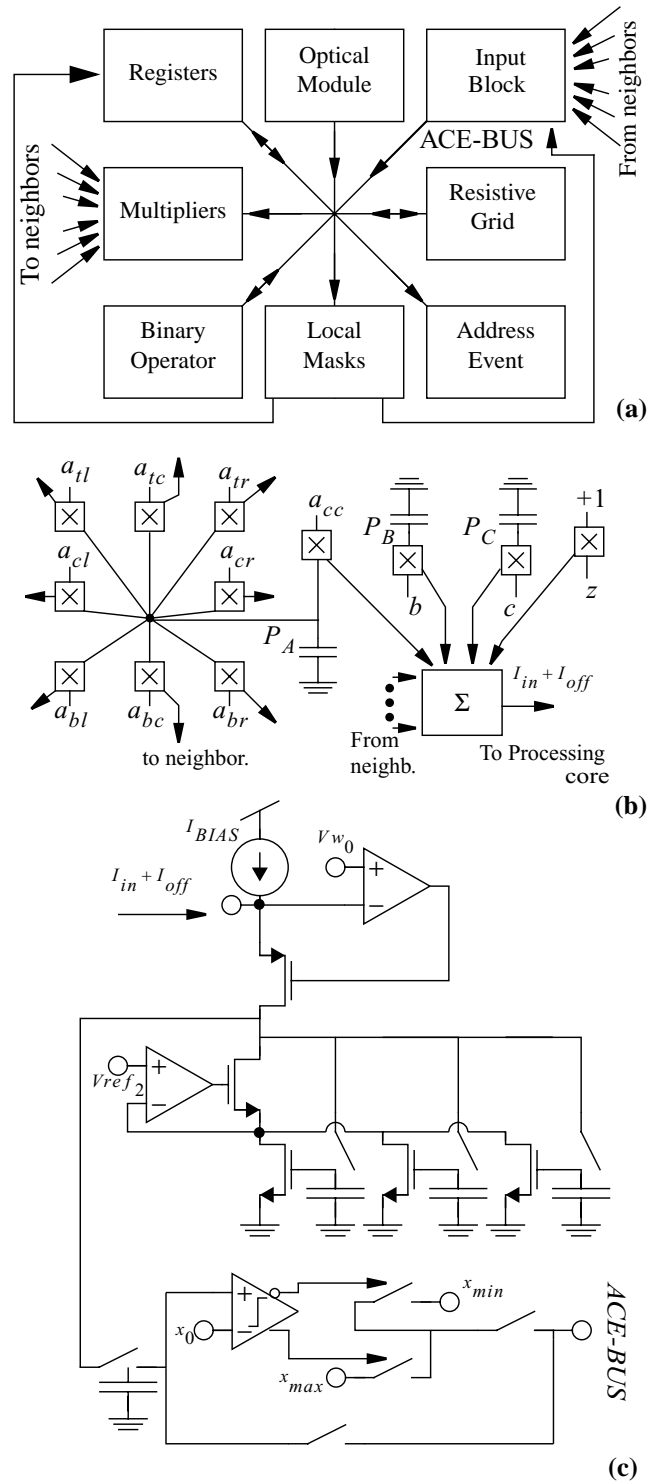


Figure 4: The Processing Element in ACE16k. a) Block Diagram. b) Bank of Multipliers. c) Current Processing Block

## 4.2. Analog Registers

The analog register, in Fig.5, stores 8 grey-scale pixel values with an equivalent resolution of 8-b. Memorization relies on a bottom-plate sampling technique, which avoids the introduction of signal dependent feedthrough error. Moreover, this technique is not sensitive to offset voltage of the required *opamp*. Hence, it is very suitable for spatial uniformity issues.

## 4.3. Optical Module

The optical input module, in Fig.6, consists of a multimode sensor in which both the physical device used as sensor and the transduction mechanism are programmable. A P-diff/N-well diode, a N-well/P-sub diode, or a P-diff/N-well/P-sub phototransistor, are available. Selection is done by using global programming instructions. Furthermore, the phototransduction scheme is also programmable. Both linear integration of the photogenerated current, or logarithmic compression sensing, are also available by proper definition of the digital instructions controlling the chip operation. This allows for the chip to be used in very different illumination conditions.

## 4.4. Binary Operator

A fully programmable 2-input 1-output binary operator has been also included. It allows for the execution of any 2-variables boolean function. Operands are acquired from the ACE-BUS, while the logic operation to be performed is defined by four global digital instruction signals -in truth-table form.

## 4.5. Local masks & Address Event Blocks

Two masks, whose use is optional -they must be enabled by the user-, have been also incorporated to the PE. The first one, called *freeze* mask, is a transient enabling-disabling mask. When active, it interrupts the flow of current from the input block to the ACE-BUS, hence, those PE's in which the content of the mask -which has been previously acquired

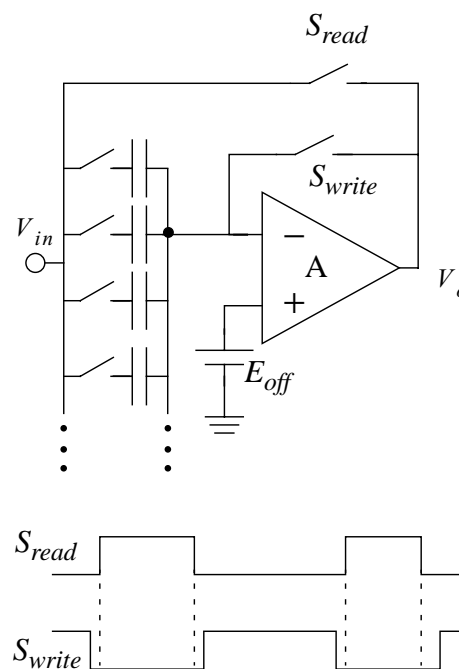


Figure 5: Sample and Hold Circuit.

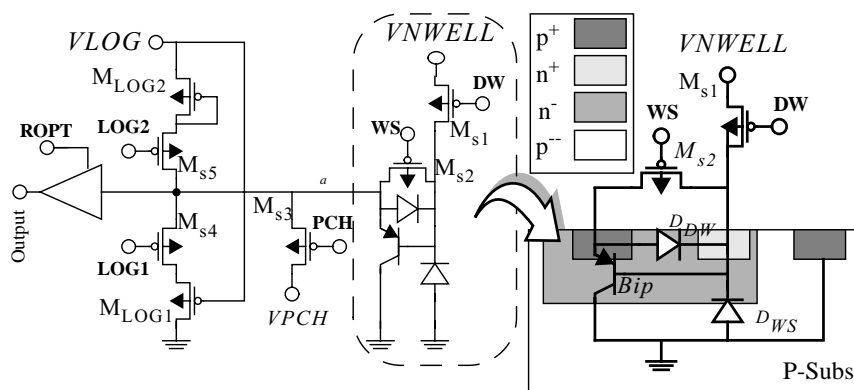


Figure 6: Representation of the Multimode Pixel

from the ACE-BUS- is +1 remain unchanged. The second one, the *writing* mask, operates in a similar way. It disables the possibility of updating any analog register. Therefore, it allows for selective updating of content of the memory.

The address event circuitry is an independent module which allows for fast downloading of sparse images. In this case, the chip provides the addresses of those pixels in which activity is detected. Active cells are those cells having a low-logic level at the ACE-BUS when the address event downloading is started.

#### 4.6. Resistive Network

A resistive grid, connecting the PE to its right and top neighbor has been included. It allows for the execution of conventional low-pass diffusion process.

### 5. I/O INTERFACE

As compared to previous analog processor implementations,  $8_{-11}$ , and regardless of the increased number of cells, one of the main improvements of ACE16k is the incorporation of a completely digital interface for system control as well as for I/O of digitized gray-scale images (see Fig. 7 (a)).

To that purpose, ACE16k incorporates 128 (one per column) Digital-to-Analog and Analog-to-Digital converters. On the one hand, the Digital-to-Analog converters required to download images in electrical form into ACE16k, are based on a resistor string and an analog multiplexer <sup>22</sup> – see Fig. 7(b).

On the other hand, the Analog-to-Digital converters – also in Fig. 7(b) –, needed to upload processed images from ACE16k, are based on a successive approximation approach <sup>22</sup>.

This choice of converter architectures provide a very good compromise in terms of area and power dissipation in this particular type of systems. On the one hand, the Digital-to-Analog converters are used as part of the successive approximation Analog-to-Digital converters by shifting  $1/2\text{LSB}$  up the levels generated by the resistor string –see Fig. 7(b). On the other hand, because the 128 converters work in parallel, a significant part of the digital circuitry needed to control the successive approximation registers can be shared by all the columns in a common peripheral block, resulting in a substantial reduction in area and power dissipation.

Finally, a self calibration process is automatically executed at the beginning of every data conversion for I/O-related fixed-pattern noise suppression.

Transferring a complete row, 128 pixels, to/from the chip requires  $1\mu\text{s}$ . Since the chip uses a two-stage pipelined architecture, the total time for image loading/uploading is  $130\mu\text{s}$ . In order to avoid undesirable digital coupling with the analog processing circuitry, image I/O and processing are normally done sequentially at different temporal windows. In most practical cases, an allocation of  $140\mu\text{s}$  for image processing is more than enough –

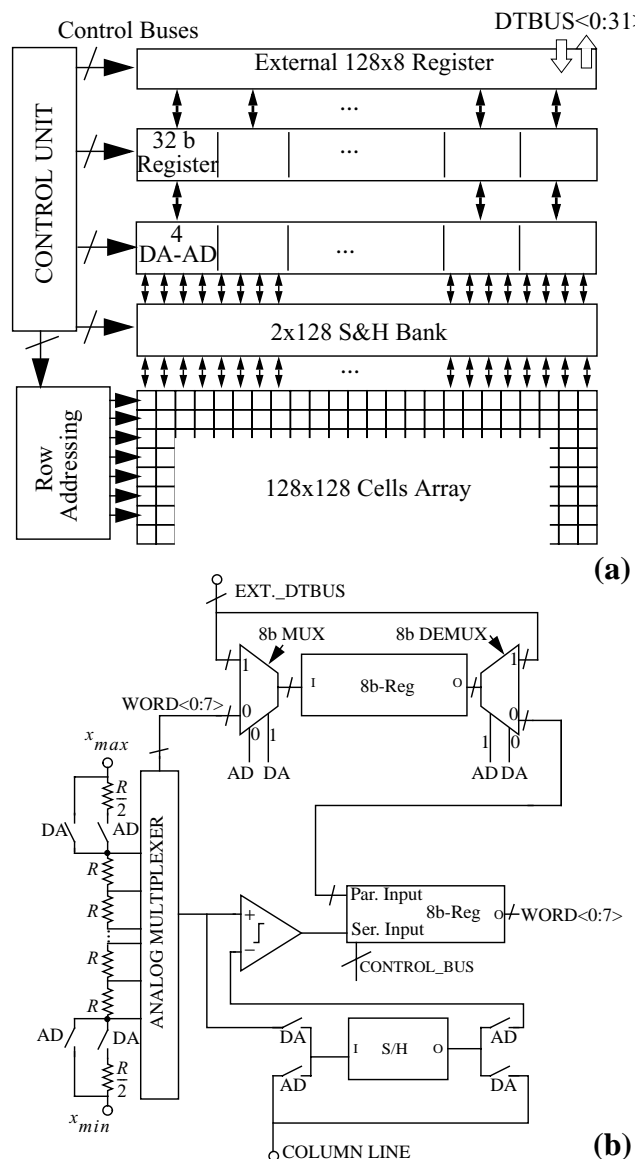


Figure 7: I/O Circuitry. a) Block Diagram b) Schematic of the I/O Block on top of one column.



around 11 basic image processing tasks can be executed within this time. With this assumption, the time required to load, process, and download a  $128 \times 128$  image is about  $400\mu\text{s}$ , reaching up to 100 VGA Frames/second.

## 6. EXPERIMENTAL RESULTS

The functional testing of the ACE16k chip, which is already running, has been performed by using a specific hardware-software platform designed at the Analogic and Neural Computing Laboratory<sup>g</sup> headed by Prof. T. Roska in Budapest.

The hosting platform, with ACE16k mounted, is displayed in Fig.8. This platform is connected, by stacking, to another board which contains the necessary logic blocks to interface the chip to/from the PCI bus.

Fig.9 shows an example of optical acquisition using ACE16k. Image size is  $128 \times 128$ . Illuminations conditions were room type, with a 60W bulb at about 1m. Exposition time was 20ms.

Finally, Fig.10 shows the results of different linear  $3 \times 3$  convolution masks executed by ACE16k, namely a low-pass operator, a Sobel operator, and a high-pass operator (in Fig.10(b) Fig.10(c), and Fig.10(d), respectively). Each convolution needs  $3.5\mu\text{s}$  for on-chip calibration of different errors, and  $1\mu\text{s}$  to reach the final steady-state.

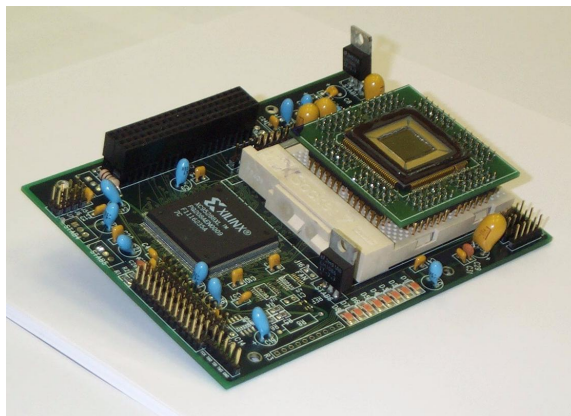


Figure 8: Hosting Platform

## 7. CONCLUSIONS

We have presented, from a high-level perspective, a new focal plane array processor chip with embedded image and programming memories. The chip, which contains close to 4 mill. transistors operating in the analog domain, is able to acquire the visual information, process the acquired images on-chip using optimized analog processing blocks, store different images at the PE level, and provide the results in a digital form. Experimental results show correct operation of the basic convolution masks used in low-level image processing. The test of the chip is in a very early stage. We expect complex vision processing algorithms, including flows, bifurcation, and conditional executions to be available for the time of the conference.



Figure 9: Hosting Platform

## 8. REFERENCES

1. P. Saffo, "Sensors: The Next Wave of InfoTech Innovation". *Institute for the Future*, 1997 Ten-Year Forecast.
2. B. Roska and F. Werblin, "Vertical Interactions Across Ten Parallel, Stacked Representations in the Mammalian Retina". *Nature*, No. 410, pp. 583-587, March 2001.

---

<sup>g</sup><http://lab.analogic.sztaki.hu/>

3. C. Koch and H. Li (Eds.), *Vision Chips, Implementing Vision Algorithms with Analog VLSI Circuits*. IEEE Press, 1995.
4. A. Moini, *Vision Chips*. Kluwer Academic Publishers, 2000.
5. T. Roska and A. Rodríguez-Vázquez (Eds.), *Towards the Visual Microprocessor*. John Wiley & Sons Ltd., 2000.
6. T. Roska and L. O. Chua: "The CNN Universal Machine: An Analogic Array Computer". *IEEE Transactions on circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 40, No. 3, pp. 163-173, March 1993.
7. R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. Carmona, P. Foldesy, A. Zarándy, P. Szolgay, T. Sziranyi and T. Roska, "A 0.8  $\mu\text{m}$  CMOS Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage". *IEEE Journal of Solid State Circuits*, Vol. 32, No. 7, pp. 1013-1026, July 1997.
8. A. Paasio, A. Dawidziuk, K. Halonen and V. Porra, "Minimum Size 0.5 $\mu\text{m}$  CMOS Programmable 48 x 48 CNN Test Chip", *Proc. of the 1997 European Conference on Circuit Theory and Design*, pp. 154-156, Budapest, Hungary, September 1997.
9. P. Kinget and M. Steyaert, "An Analog Parallel Array Processor for Real-Time Sensor Signal Processing". *Proc. of the IEEE International Solid-State Circuits Conference*. pp. 92-93. San Francisco, CA, USA, February 1996.
10. P. Dudek, *A Programmable Focal-Plane Analogue Processor Array*. Ph. D. Dissertation, University of Manchester Institute of Science and Technology, May 2000.
11. G. Liñán, S. Espejo, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "ACE4k: An Analog I/O 64x64 visual microprocessor chip with 7-bit accuracy". *Int. Journal of Circuit Theory and Applications*, Vol. 30, pp. 89-116, March 2002.
12. T. Bernard, B. Y. Zavidovique, and F. J. Devos, "A Programmable Artificial Retina". *IEEE J. of Solid State Circuits*, Vol. 28, No. 7, pp. 789-798, July 1993.
13. J. C. Gealow and C. G. Sodini, "A Pixel-Parallel Image Processor Using Logic Pitch-Matched to Dynamic Memory". *IEEE J. of Solid State Circuits*, pp. 831- 839, Vol. 34, No. 6, June 1999.
14. M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii, "A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing". *Proc. of the ISSCC*, TP. 12.2, pp. 206-207, 1999.
15. N. Yamashita et al. "A 3.84 GIPS Integrated Memory Array Processor with 64 Processing Elements and a 2-Mb RAM". *IEEE J. of Solid State Circuits*, Vol. 29, No. 11, pp. 1336-1343, Nov. 1994.
16. R. Etienne-Cummings, Z. Kevork Kalayjian, and D. Cai, "A Programmable Focal Plane MIMD Image Processor Chip". *IEEE J. of Solid State Circuits*, Vol. 36, No. 1, pp. 64-73, Jan. 2001.
17. Texas Instruments on the web: <http://www.ti.com>.
18. B. Jahne, H. Haubecker and P. Geibler (Eds.), *Handbook of Computer Vision and Applications*. Academic Press, London, ISBN 0-12-379771, 1999.
19. T. Roska, L. Kék, L. Nemes, Á. Zarándy, M. Brendel, *CSL - CNN Software Library - Version 7.2*. Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.
20. A. Rodríguez-Vázquez, R. Domínguez-Castro, F. Medeiro, J.L. Huertas and M. Delgado-Restituto, "High-Resolution CMOS Current Comparators: Design and Applications to Current-Mode Function Generation". *Analog Integrated Circuits and Signal Processing*, Vol. 7, pp. 149-165, March 1995.
21. S. Espejo, R. Carmona, R. Domínguez-Castro and A. Rodríguez-Vázquez, "A VLSI-Oriented Continuous-Time CNN Model". *International Journal of Circuit Theory and Applications*. Vol.24, pp.341-356, May-June 1996.
22. B. Razavi, *Principles of Data Conversion System Design*. IEEE Press, New York, 1995, ISBN: 0-7803-1093-4

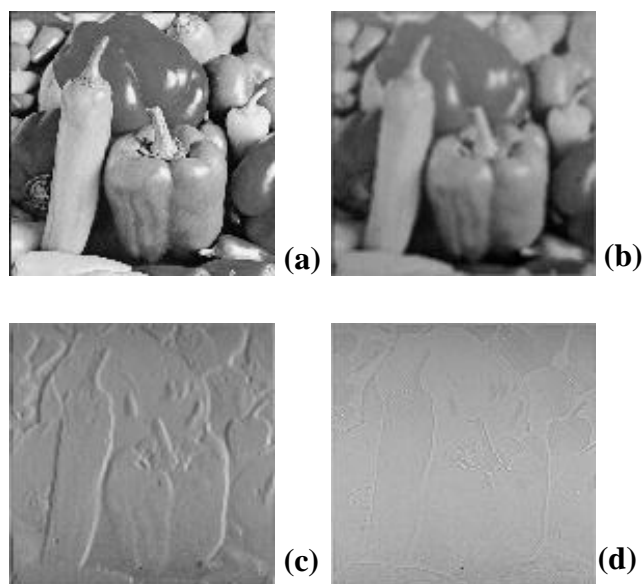


Figure 10: Results of Linear Convolutions